# Architecture Blueprint: Air Quality Hybrid Pipeline

## 1. Executive Summary

This document outlines the cloud architecture for the Hybrid Air Quality Monitoring System. The system integrates historical batch data (2004) with real-time streaming data (2025) to provide predictive analytics on environmental pollutants (CO, NO2, O3). The architecture is designed to be **serverless**, **cost-optimized**, and **scalable**.

## 2. Architecture Diagram Description

The pipeline follows the "Lambda Architecture" pattern, handling both batch and stream layers:

1. **Ingestion Layer:**
   - **Batch:** Raw CSV files sourced from Kaggle are uploaded to **Google Cloud Storage (GCS)**.
   - **Streaming:** A **Cloud Scheduler** trigger invokes a **Cloud Function (Gen 2)** every 15 minutes. The function polls the Open-Meteo API.
2. **Messaging Layer:**
   - The Cloud Function publishes normalized JSON payloads to **Pub/Sub**.
3. **Storage Layer:**
   - **Raw Data Lake:** GCS bucket stores the immutable historical CSVs.
   - **Data Warehouse: BigQuery** serves as the central repository.
     - *Batch Table:* Ingested from GCS, partitioned by Day.
     - *Streaming Table:* Ingested directly from Pub/Sub via BigQuery Subscription.
4. **Analytics & ML Layer:**
   - **BigQuery ML (BQML):** A Linear Regression model is trained on the Batch table and deployed to score incoming data in the Streaming table.
   - **Views:** Virtual tables handle unit conversion, time-smoothing, and joining historical baselines with live data.
5. **Visualization Layer:**
   - **Looker Studio:** Connects to BigQuery Views to render Executive KPIs and Time-Series dashboards.

## 3. Security & Compliance

- **Authentication:** The Cloud Function runs with a specific Service Account (`runtime service account`).
- **Authorization:** IAM roles are restricted. The Pub/Sub Service Agent is granted only `bigquery.dataEditor` on the specific dataset.

- **Secret Management:** API Keys (if required in future iterations) are injected via **Google Secret Manager** or secure Environment Variables, never hardcoded.
- **Network:** All services communicate over Google's internal backbone.

# 4. Cost Control Strategy

- **Serverless:** Cloud Functions scale to zero; costs are incurred only during the ~500ms execution time.
- **BigQuery Storage:** Historical data utilizes Long-Term Storage pricing (50% discount) automatically after 90 days.
- **Partitioning:** The historical table is partitioned by Date, reducing query costs by scanning only relevant partitions.
- **Quotas:** A `max_bytes_billed` limit is recommended for the BigQuery project to prevent accidental over-spending.